Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'X509_NAME_print_ex_fp.3ossl' command

$ man X509_NAME_print_ex_fp.3ossl

X509_NAME_PRINT_EX(3ossl) OpenSSL X509_NAME_PRINT_EX(3ossl)

NAME

X509_NAME_print_ex, X509_NAME_print_ex_fp, X509_NAME_print,

X509_NAME_oneline - X509_NAME printing routines

SYNOPSIS

```
#include <openssl/x509.h>
int X509_NAME_print_ex(BIO *out, const X509_NAME *nm,
                       int indent, unsigned long flags);
int X509_NAME_print_ex_fp(FILE *fp, const X509_NAME *nm,
                          int indent, unsigned long flags);
char *X509_NAME_oneline(const X509_NAME *a, char *buf, int size);
int X509_NAME_print(BIO *bp, const X509_NAME *name, int obase);
```

DESCRIPTION

X509_NAME_print_ex() prints a human readable version of nm to BIO out. Each line (for multiline formats) is indented by indent spaces. The output format can be extensively customised by use of the flags parameter.

X509_NAME_print_ex_fp() is identical to X509_NAME_print_ex() except the output is written to FILE pointer fp.

X509_NAME_oneline() prints an ASCII version of a to buf. This supports multi-valued RDNs and escapes / and + characters in values. If buf is NULL then a buffer is dynamically allocated and returned, and size is ignored. Otherwise, at most size bytes will be written, including the

ending '\0', and buf is returned.

X509_NAME_print() prints out name to bp indenting each line by obase characters. Multiple lines are used if the output (including indent) exceeds 80 characters.

NOTES

The functions X509_NAME_oneline() and X509_NAME_print() produce a non standard output form, they don't handle multi-character fields and have various quirks and inconsistencies. Their use is strongly discouraged in new applications and they could be deprecated in a future release.

Although there are a large number of possible flags for most purposes XN_FLAG_ONELINE, XN_FLAG_MULTILINE or XN_FLAG_RFC2253 will suffice. As noted on the ASN1_STRING_print_ex(3) manual page for UTF8 terminals the ASN1_STRFLGS_ESC_MSB should be unset: so for example XN_FLAG_ONELINE & ~ASN1_STRFLGS_ESC_MSB would be used.

The complete set of the flags supported by X509_NAME_print_ex() is listed below.

Several options can be ored together.

The options XN_FLAG_SEP_COMMA_PLUS, XN_FLAG_SEP_CPLUS_SPC, XN_FLAG_SEP_SPLUS_SPC and XN_FLAG_SEP_MULTILINE determine the field separators to use. Two distinct separators are used between distinct RelativeDistinguishedName components and separate values in the same RDN for a multi-valued RDN. Multi-valued RDNs are currently very rare so the second separator will hardly ever be used.

XN_FLAG_SEP_COMMA_PLUS uses comma and plus as separators.

XN_FLAG_SEP_CPLUS_SPC uses comma and plus with spaces: this is more readable that plain comma and plus. XN_FLAG_SEP_SPLUS_SPC uses spaced semicolon and plus. XN_FLAG_SEP_MULTILINE uses spaced newline and plus respectively.

If XN_FLAG_DN_REV is set the whole DN is printed in reversed order.

The fields XN_FLAG_FN_SN, XN_FLAG_FN_LN, XN_FLAG_FN_OID, XN_FLAG_FN_NONE determine how a field name is displayed. It will use the short name (e.g. CN) the long name (e.g. commonName) always use OID numerical form (normally OIDs are only used if the field name is not

recognised) and no field name respectively.

If XN_FLAG_SPC_EQ is set then spaces will be placed around the '='

character separating field names and values.

If XN_FLAG_DUMP_UNKNOWN_FIELDS is set then the encoding of unknown

fields is printed instead of the values.

If XN_FLAG_FN_ALIGN is set then field names are padded to 20

characters: this is only of use for multiline format.

Additionally all the options supported by ASN1_STRING_print_ex() can be

used to control how each field value is displayed.

In addition a number options can be set for commonly used formats.

XN_FLAG_RFC2253 sets options which produce an output compatible with

RFC2253. It is equivalent to:

"ASN1_STRFLGS_RFC2253 | XN_FLAG_SEP_COMMA_PLUS | XN_FLAG_DN_REV

| XN_FLAG_FN_SN | XN_FLAG_DUMP_UNKNOWN_FIELDS"

XN_FLAG_ONELINE is a more readable one line format which is the same

as:

"ASN1_STRFLGS_RFC2253 | ASN1_STRFLGS_ESC_QUOTE | XN_FLAG_SEP_CPLUS_SPC

| XN_FLAG_SPC_EQ | XN_FLAG_FN_SN"

XN_FLAG_MULTILINE is a multiline format which is the same as:

"ASN1_STRFLGS_ESC_CTRL | ASN1_STRFLGS_ESC_MSB | XN_FLAG_SEP_MULTILINE

| XN_FLAG_SPC_EQ | XN_FLAG_FN_LN | XN_FLAG_FN_ALIGN"

XN_FLAG_COMPAT uses a format identical to X509_NAME_print(): in fact it

calls X509_NAME_print() internally.

RETURN VALUES

X509_NAME_oneline() returns a valid string on success or NULL on error.

X509_NAME_print() returns 1 on success or 0 on error.

X509_NAME_print_ex() and X509_NAME_print_ex_fp() return 1 on success or

0 on error if the XN_FLAG_COMPAT is set, which is the same as

X509_NAME_print(). Otherwise, it returns -1 on error or other values

on success.

SEE ALSO

ASN1_STRING_print_ex(3)

COPYRIGHT